

Course Material

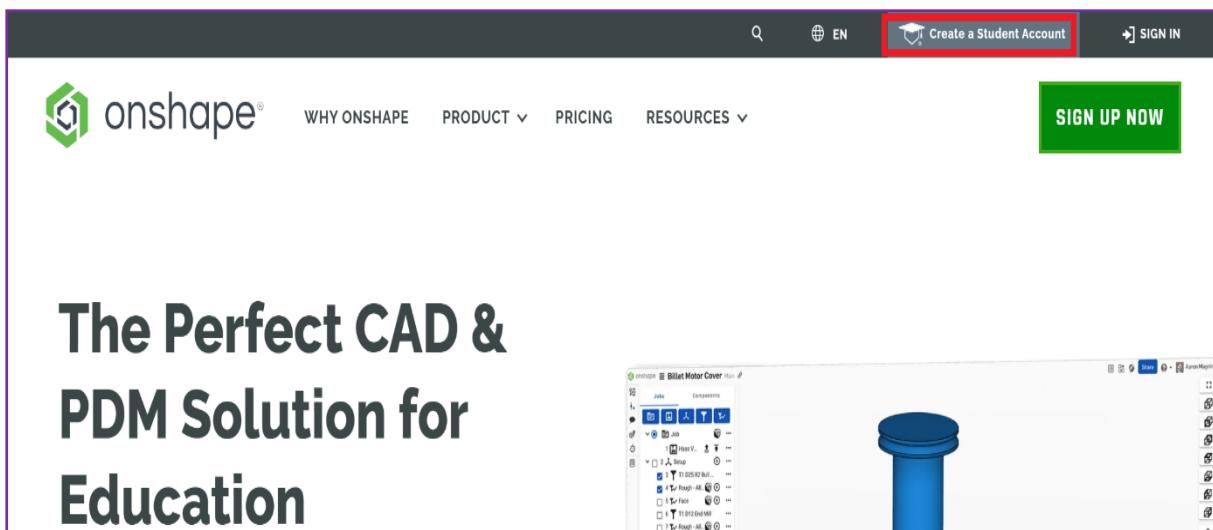
Onshape

Table of Contents

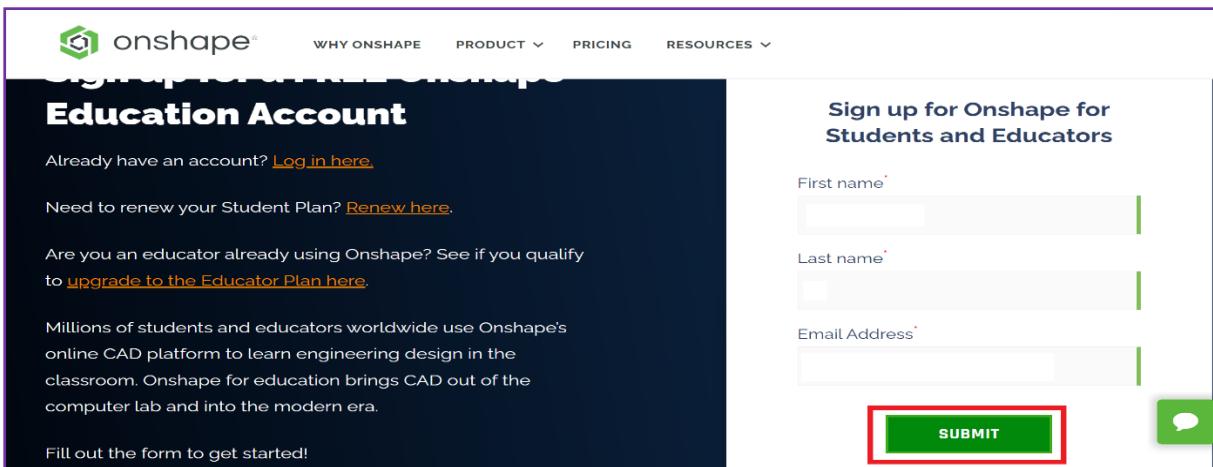
Installation.....	1
Basic Navigation Controls	3
Onshape – Variable Table.....	3
Onshape – Variables.....	6
Onshape – Feature Studio	8

Installation

1. Go to <https://www.onshape.com/en/>
2. Click on "Create a Student Account" → Select "Windows Installer (.exe)".



3. Enter your student credentials and complete the sign-up process.



Sign up for Onshape for Students and Educators

First name:

Last name:

Email Address:

SUBMIT

This will help us personalize your experience.

Student
I am currently attending school. 

Educator
I'm a teacher in a school or team mentor.

"Onshape's cloud native architecture allows for next-level collaboration in team and classroom environments, and makes learning CAD easy for any student, regardless of experience."

The school level I attend or teach is:

Grade School / K-12 (Ages <18) College / University (Ages 18+)

NEXT 

START OVER

Complete your sign up for Onshape Student

Almost done!

School Name * Sri Eshwar College of Engineering

School website URL * <https://sece.ac.in/>

Graduation Year * 2027

Area of study / Degree * Mechanical Engineering

What are you using Onshape for? * Learning

I am a Qualified Educational Institution, an Educator, a Student, a team mentor in a qualified student competition, or I have parental responsibility for a Student or I have the consent from the person with parental responsibility.

I agree to use this plan for classroom instruction, student learning projects, school clubs/organizations, and/or academic research. I will not use this plan for government, commercial, or other organizational use.

I agree to these terms as well as Onshape's [Terms of Use](#) and [Privacy Policy](#) and confirm that the above information is accurate and truthful.

COMPLETE SIGN UP

4. After sign-up, you will be redirected to the Onshape dashboard. Click on "Create a document".

Owned by me

- Owned by me
 - Recently opened
 - Created by me
 - Shared with me
- Labels
- Public
- Trash

Getting started with Onshape

Design it together with a document

In Onshape, **documents** are containers for parts, assemblies, drawings, and other related data. Each one lives within its own tab in the document.

Onshape Quickstart **Create a document** Take a self-paced course

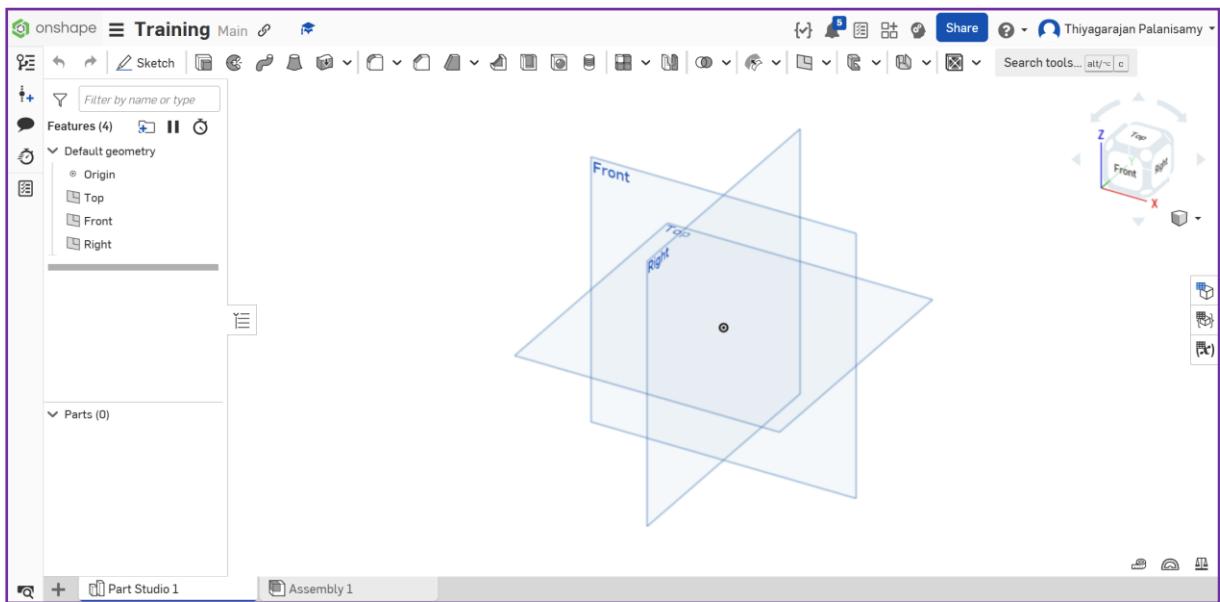
Last opened by me

Training Learnlike

Subscription: Student

© 2013 - 2025, PTC Inc. All Rights Reserved. Terms & Privacy (L19552501.10b8374ce335)

5. Onshape web application will launch with the given project name.

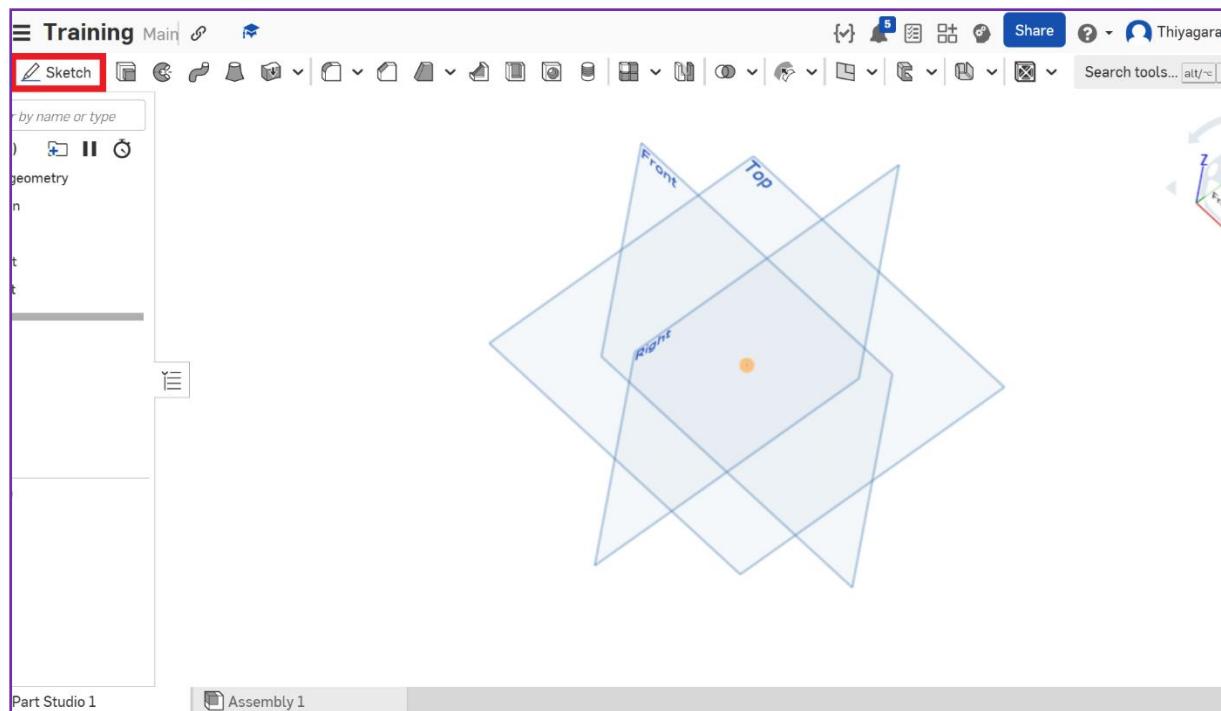


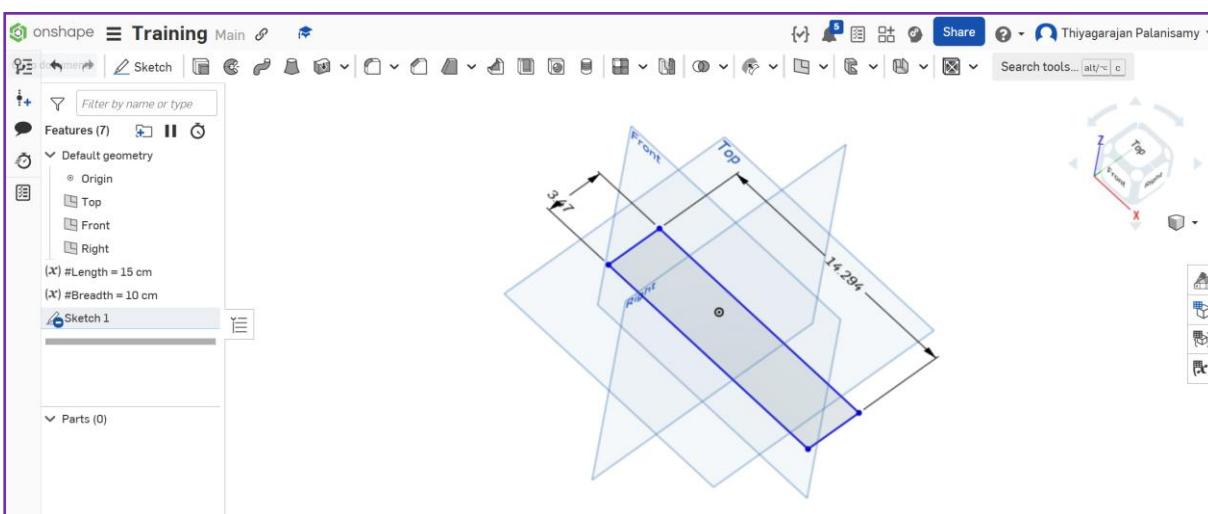
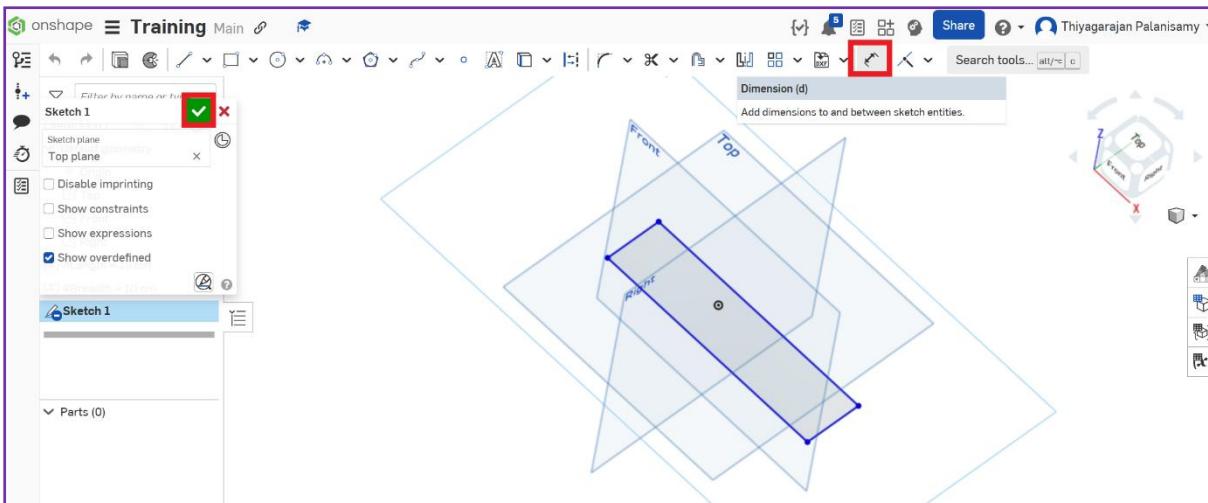
Basic Navigation Controls

- **Zoom** → Use the **mouse scroll** to zoom in and out.
- **Free Rotate** → Hold **right mouse button** to freely rotate the view.
- **Pan** → Hold **middle mouse click** to pan across the workspace.

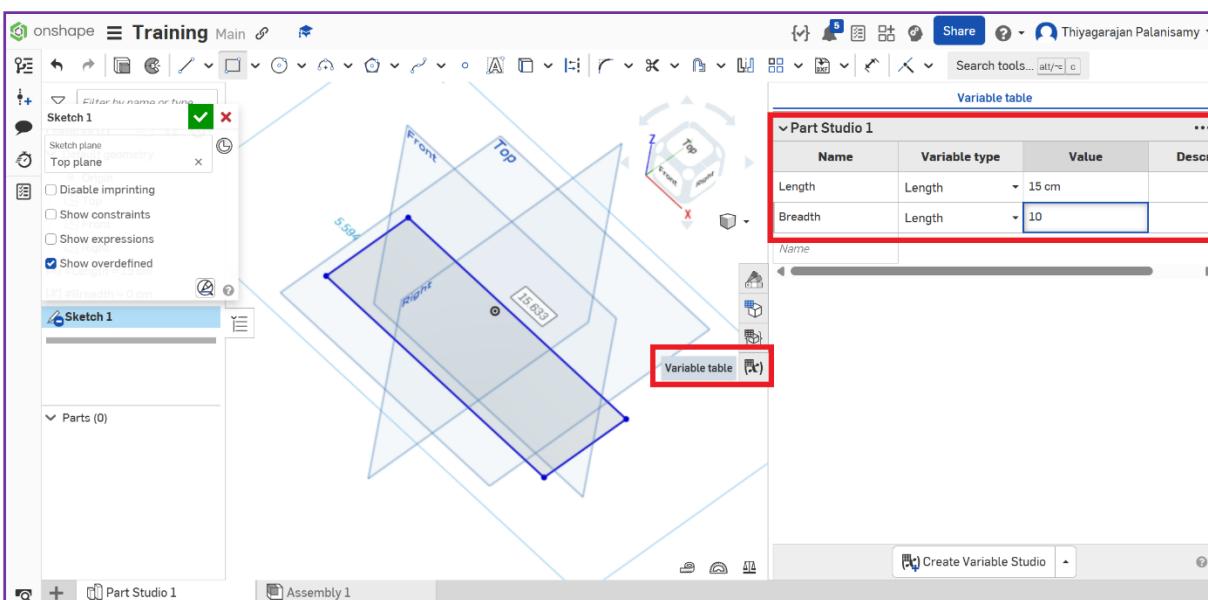
Onshape – Variable Table

1. Click on “**Sketch**” and choose a plane to draw any shape. Add dimensions to the sketch and click on the “**tick mark**” to complete the sketch.

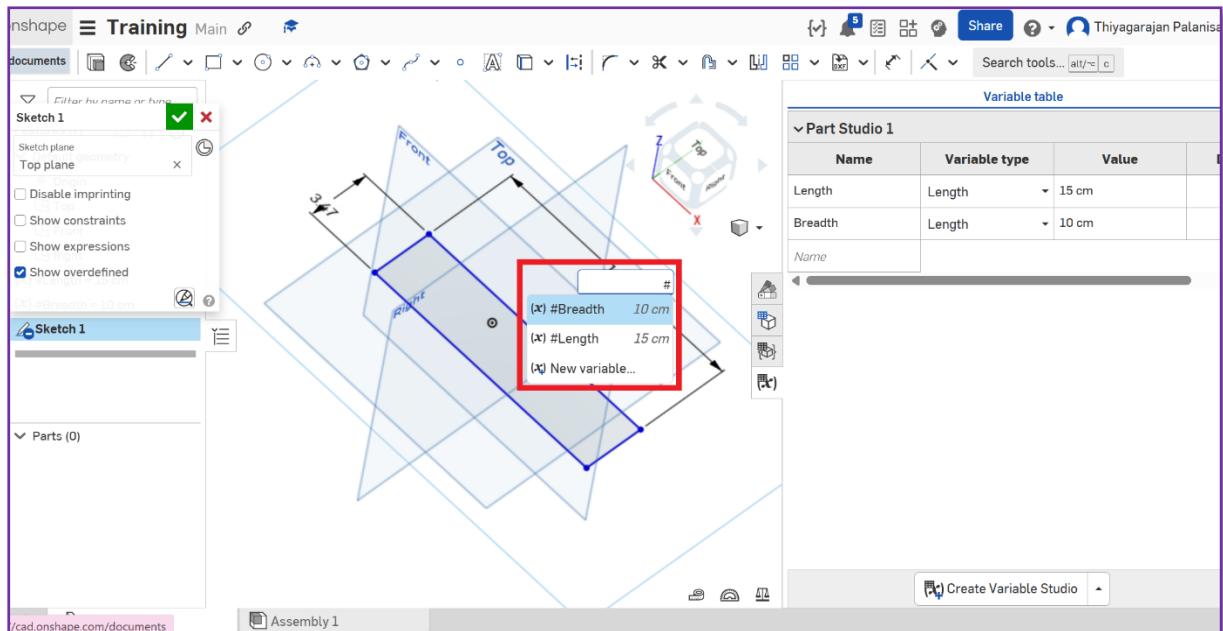




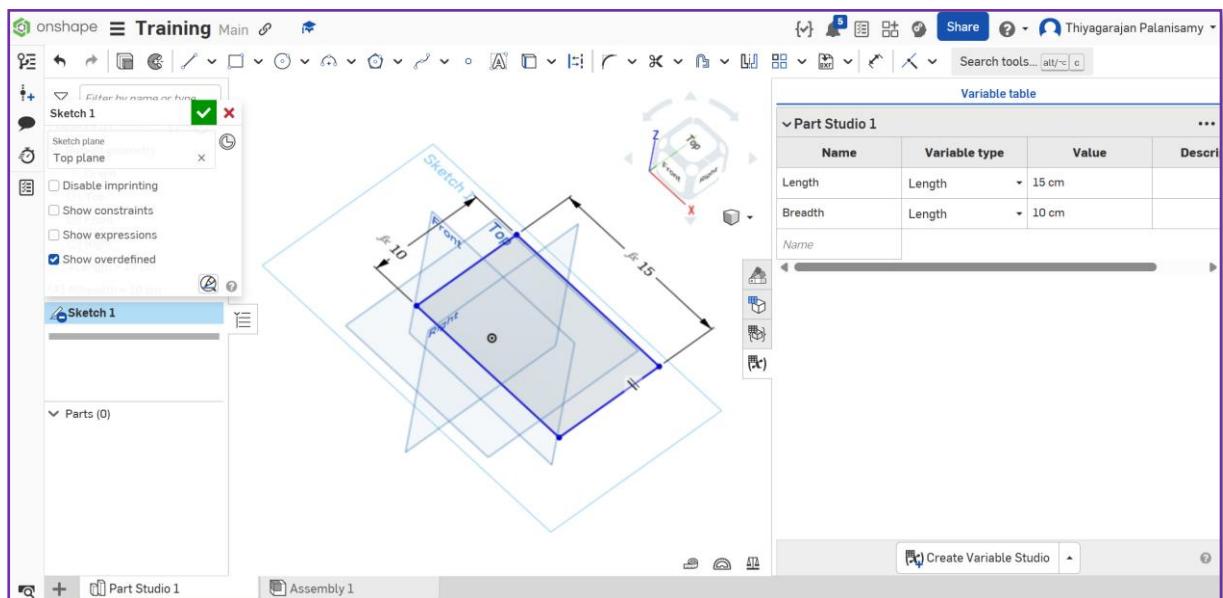
2. Create a “**Variable table**” with two entries, **Length** and **Breadth**, and define the dimensions for these variables.

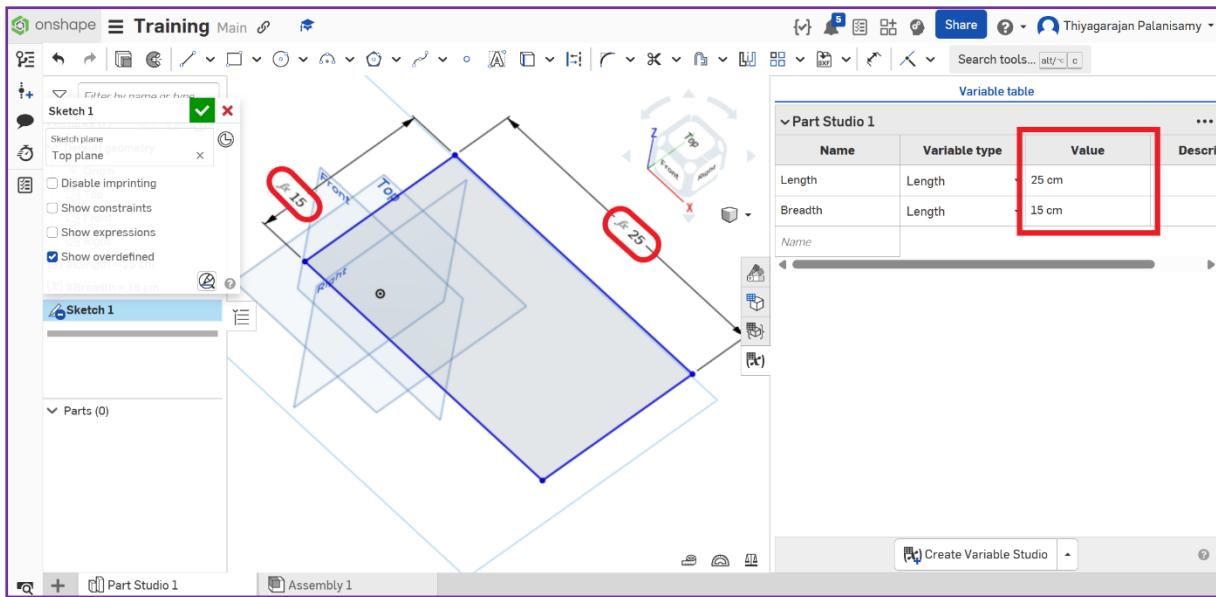


3. Assign the variable table to the sketch dimensions using # as a prefix. Double-click on one of the dimensions, type #, and choose a parameter from the variable table.

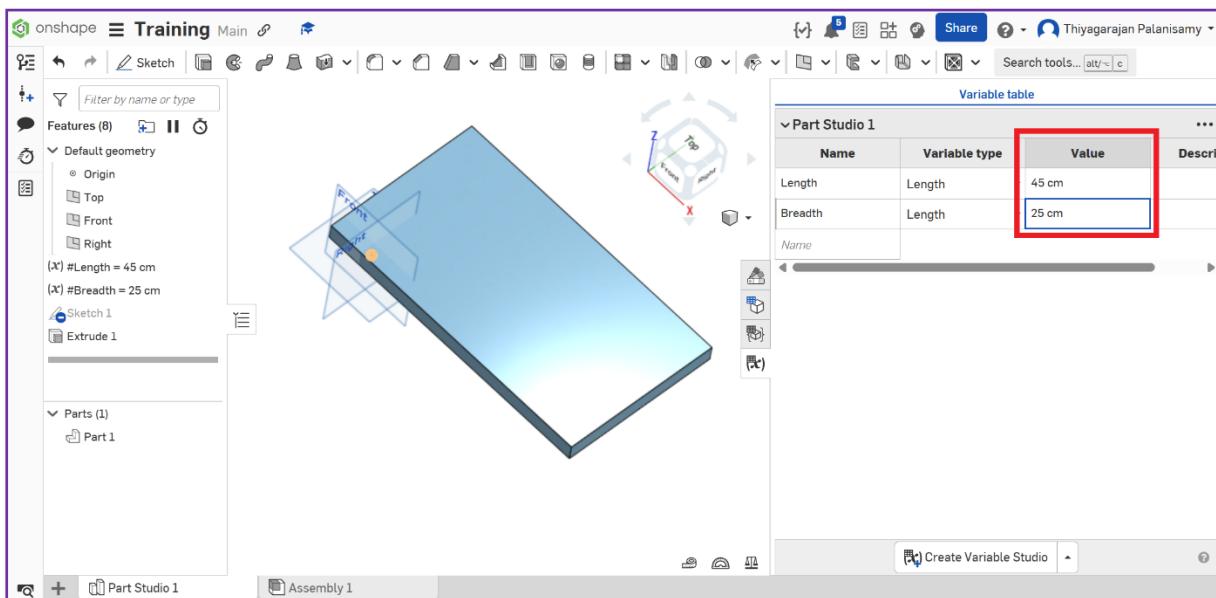


4. Once variables are assigned, a symbol fx will appear next to the dimensions. Now, change the dimension values in the variable table and verify if the sketch updates accordingly



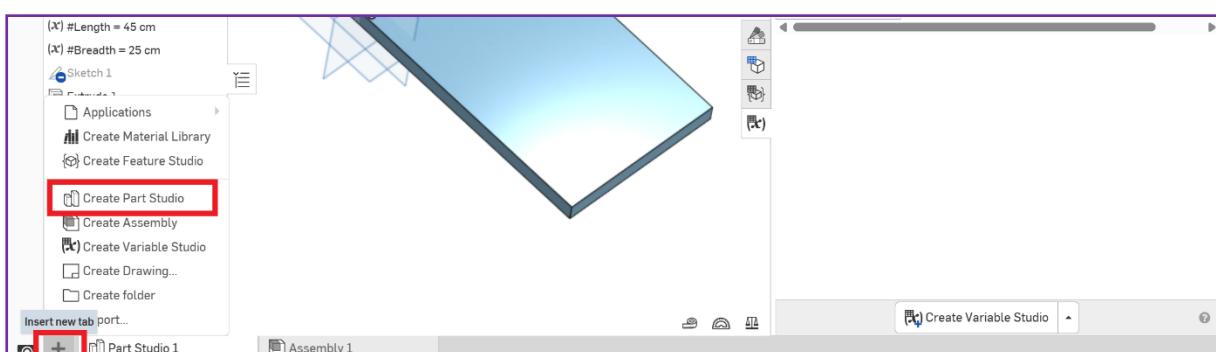


5. The table continues to work even after the sketch is converted into a solid using extrude, revolve, etc.

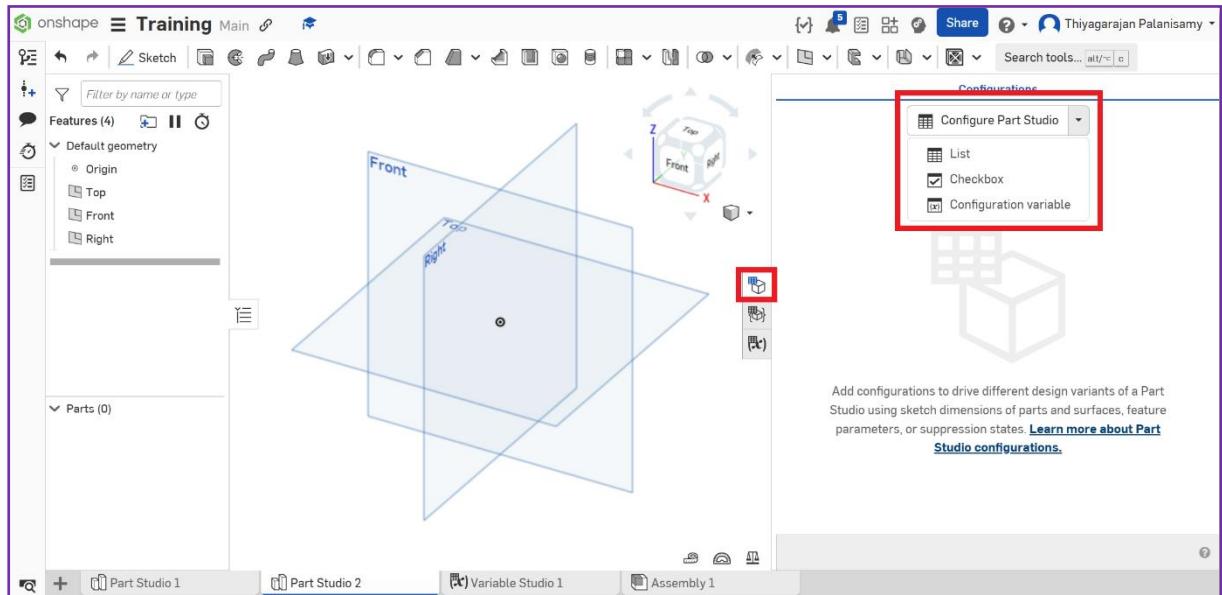


Onshape – Variables

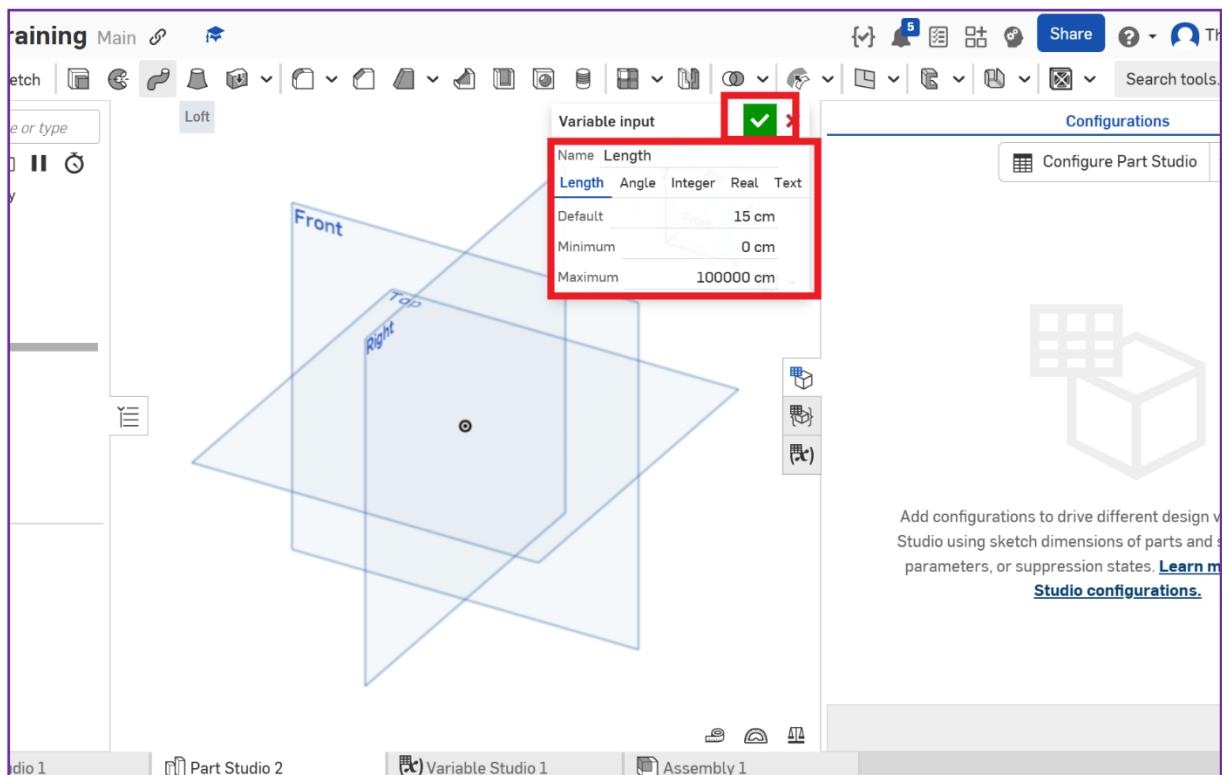
1. Click on “Insert new tab” and choose “Create Part Studio”.



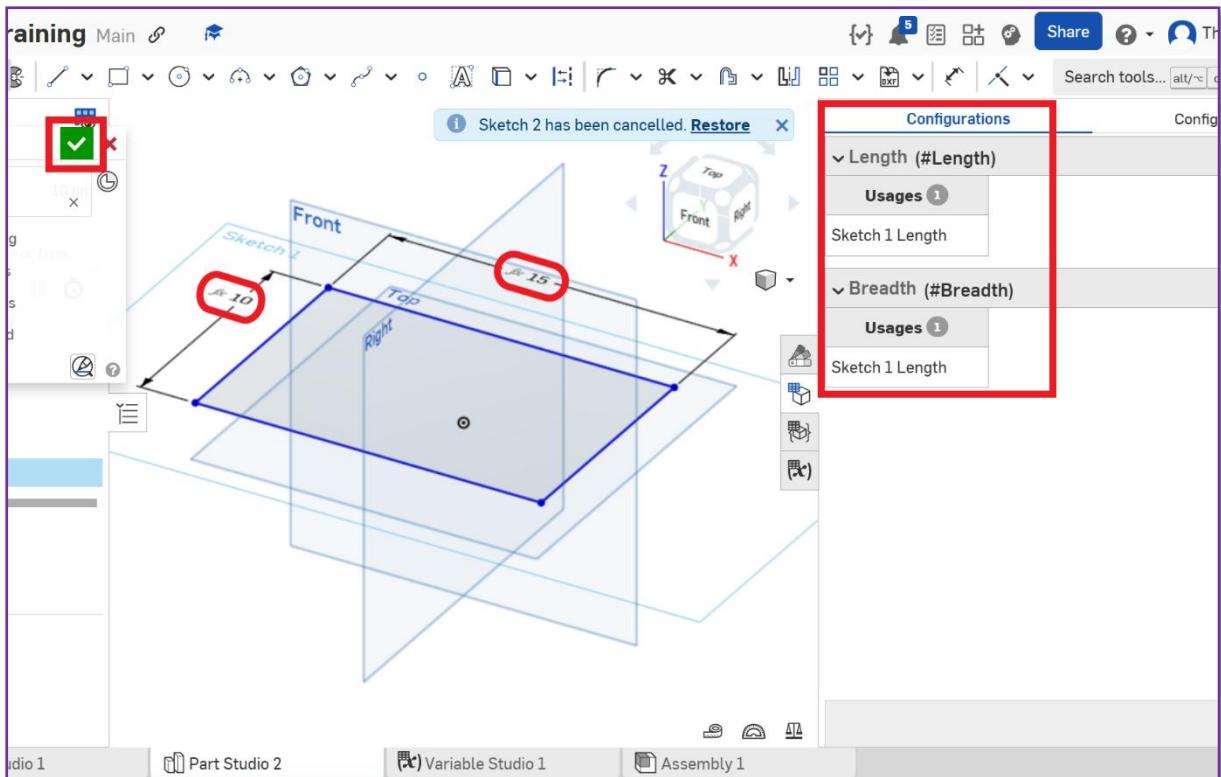
2. Click on “Configuration panel” and under “Configuration Part Studio” choose “Configuration variable”.



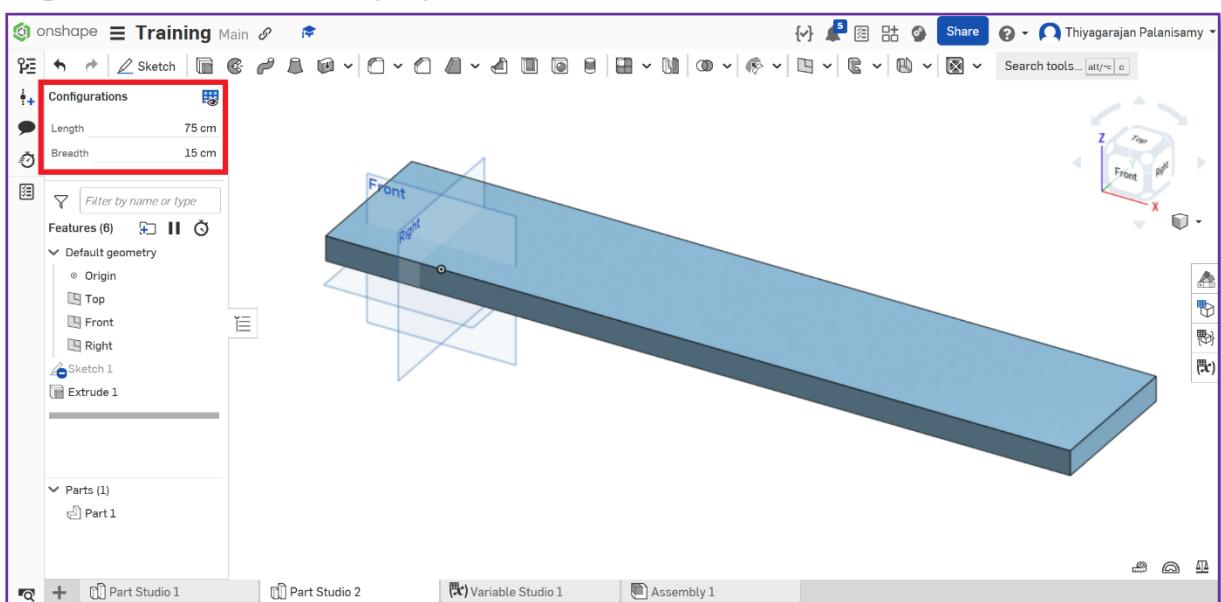
3. Create two variables named Length and Breadth. You can change the default values for both variables as needed.



4. Draw a sketch and assign the variables to the dimensions using the # prefix, as demonstrated in the previous topic in Step 3.

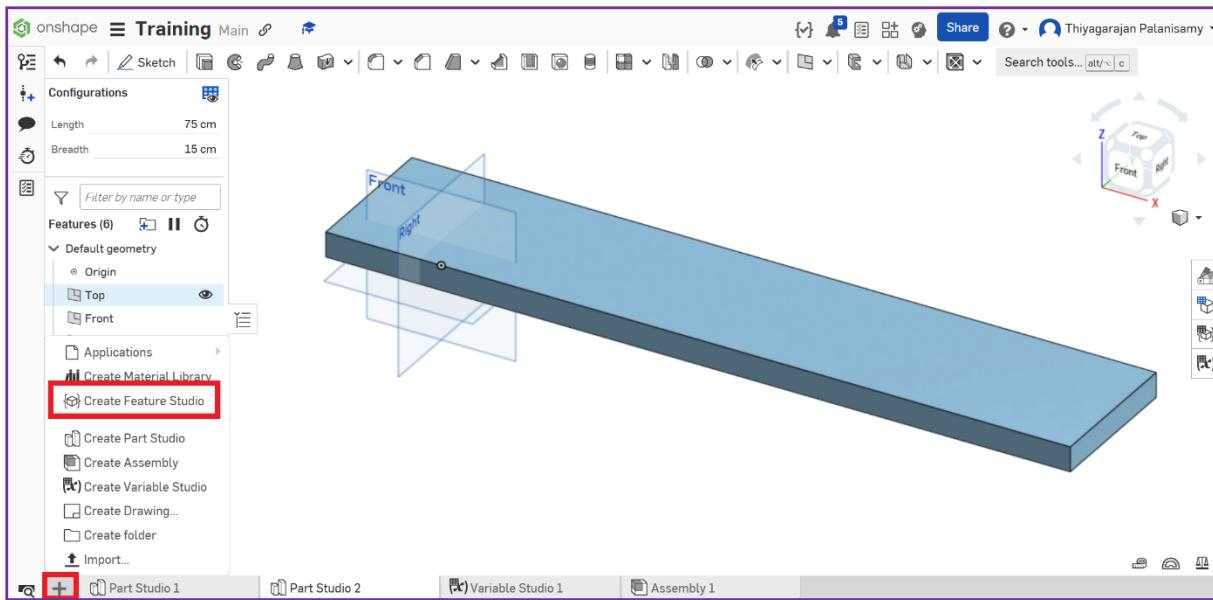


5. Now, the variables can be directly modified using an input form on the left panel, instead of changing them in the variable table.

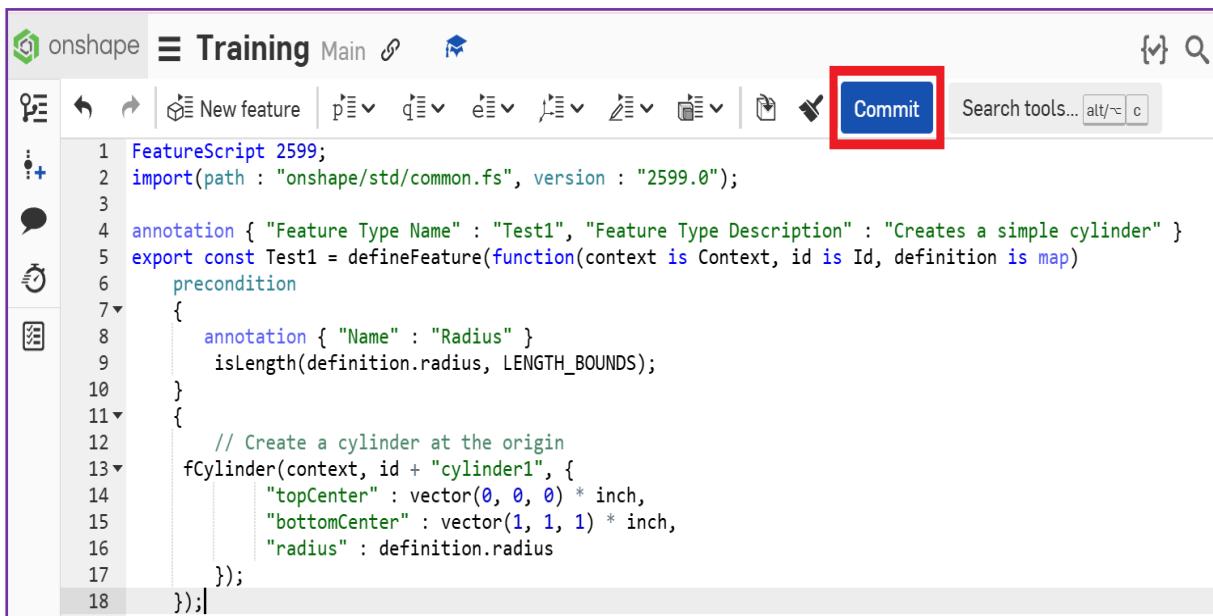


Onshape – Feature Studio

1. Click on “Insert new tab” and choose “Create Feature Studio”.



2. Copy the code from the shared document [Onshape Featurescript Samplecode Cylinder.txt](#) and paste it into the Onshape Feature Studio editor. Click on “Commit” to apply the code.



The screenshot shows the Feature Studio editor with the 'Training' project. The code entered is:

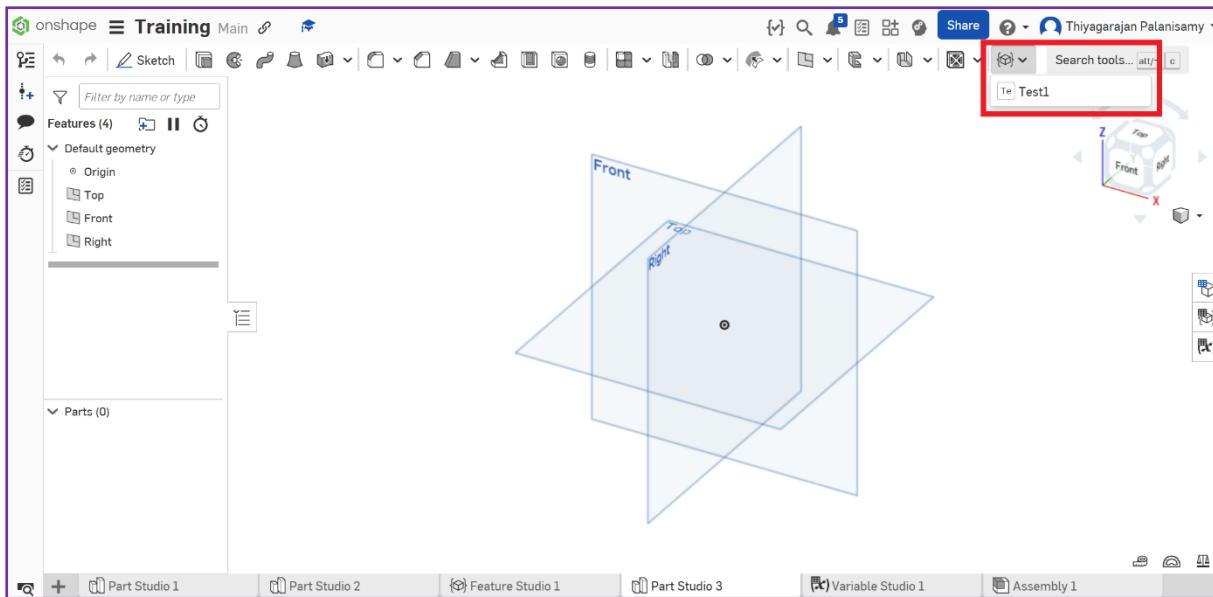
```

1 FeatureScript 2599;
2 import(path : "onshape/std/common.fs", version : "2599.0");
3
4 annotation { "Feature Type Name" : "Test1", "Feature Type Description" : "Creates a simple cylinder" }
5 export const Test1 = defineFeature(function(context is Context, id is Id, definition is map)
6     precondition
7     {
8         annotation { "Name" : "Radius" }
9             isLength(definition.radius, LENGTH_BOUNDS);
10    }
11    {
12        // Create a cylinder at the origin
13        fCylinder(context, id + "cylinder1", {
14            "topCenter" : vector(0, 0, 0) * inch,
15            "bottomCenter" : vector(1, 1, 1) * inch,
16            "radius" : definition.radius
17        });
18    });

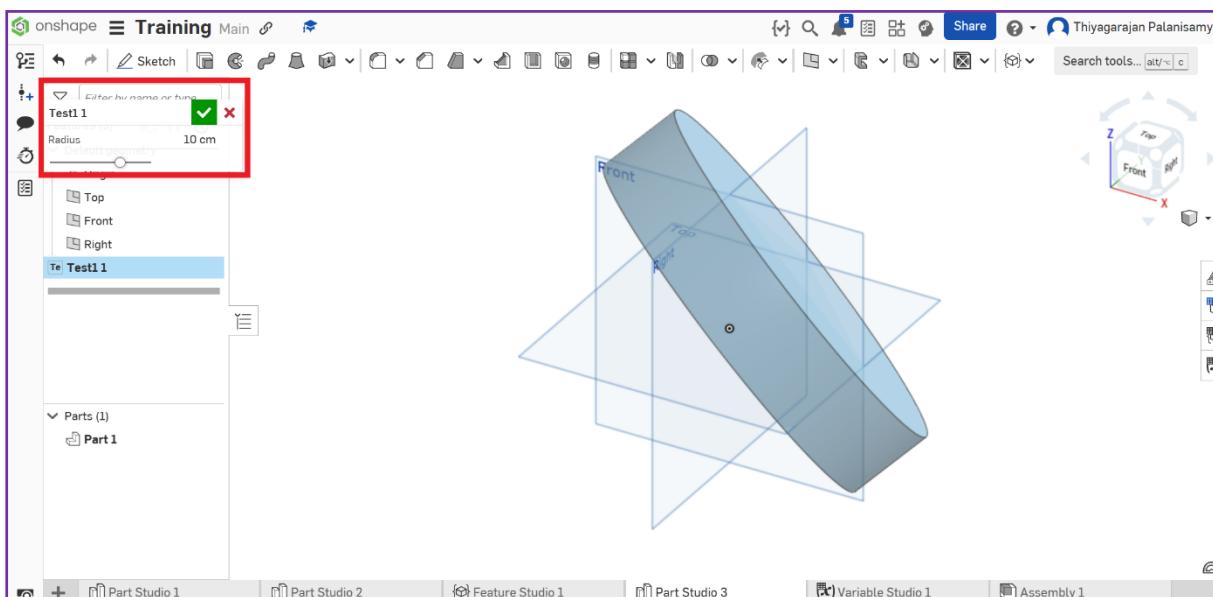
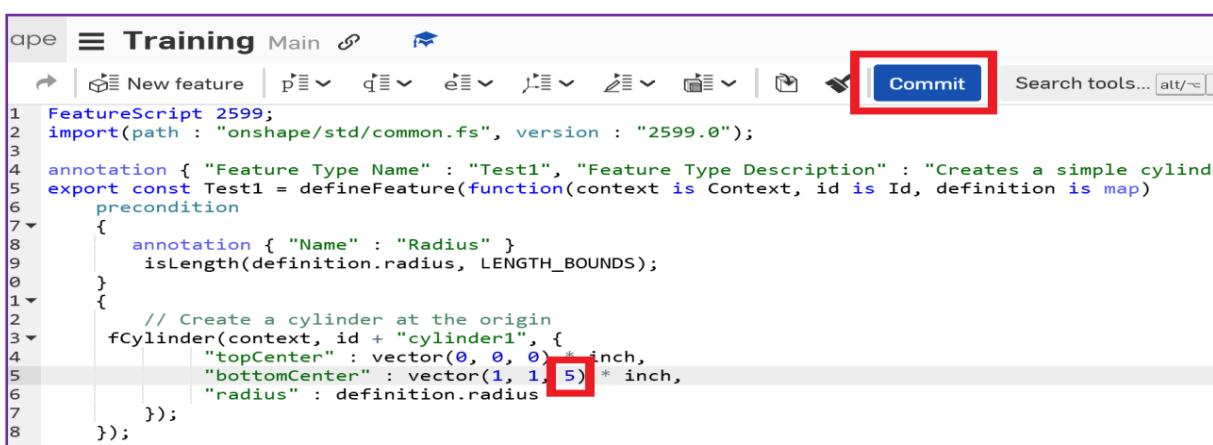
```

A red box highlights the 'Commit' button in the top right corner of the editor window.

3. Create a new Part Studio tab, click on the "Custom Features" dropdown, and select "Test1," which corresponds to the code entered in the Feature Studio.



4. A cylinder will be created at the vertex (0,0,0) using the Feature Studio code. The radius of the cylinder can be defined, and its depth can be adjusted by modifying the Z vertex coordinate in the code.

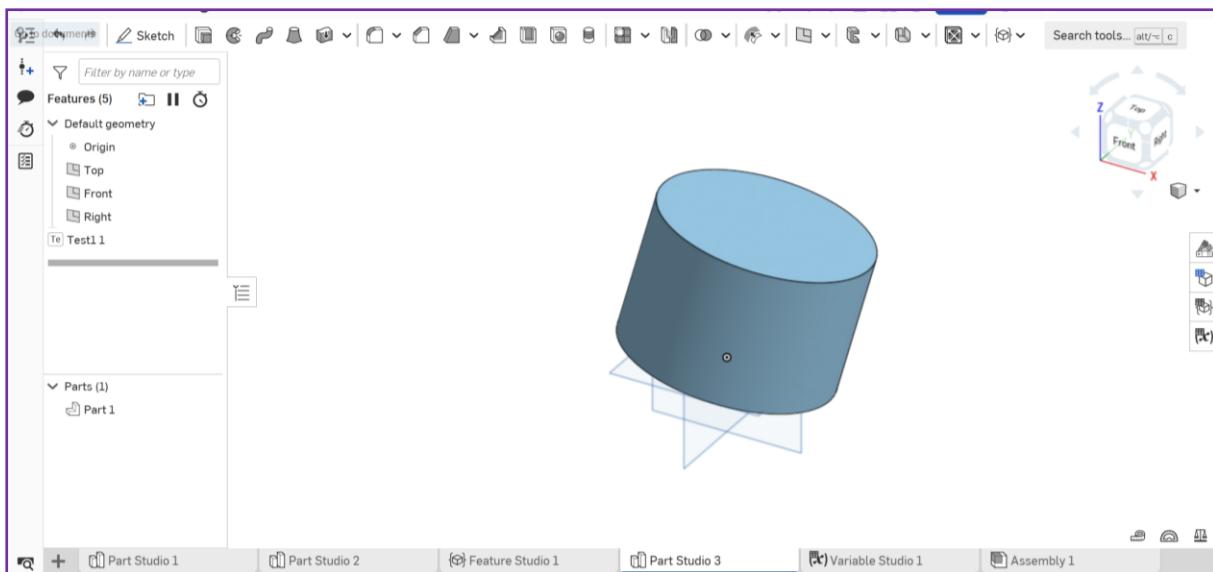



```

1 FeatureScript 2599;
2 import(path : "onshape/std/common.fs", version : "2599.0");
3
4 annotation { "Feature Type Name" : "Test1", "Feature Type Description" : "Creates a simple cylinder" };
5 export const Test1 = defineFeature(function(context is Context, id is Id, definition is map)
6   precondition
7   {
8     annotation { "Name" : "Radius" }
9     isLength(definition.radius, LENGTH_BOUNDS);
10   }
11   {
12     // Create a cylinder at the origin
13     fCylinder(context, id + "cylinder1", {
14       "topCenter" : vector(0, 0, 0) * inch,
15       "bottomCenter" : vector(1, 1, 5) * inch,
16       "radius" : definition.radius
17     });
18   });

```

The FeatureScript code defines a feature named 'Test1' that creates a cylinder at the origin (0,0,0). The cylinder has a radius of 10 cm (10 * inch) and a depth of 5 inches. The 'Commit' button in the toolbar is highlighted with a red box.



5. Apply the same method to the next two codes available in the shared drive:
 - [Onshape_Featurescript_Samplecode_Cuboid.txt](#)
 - [Onshape_Featurescript_Samplecode_Cylinderonaplane.txt](#)

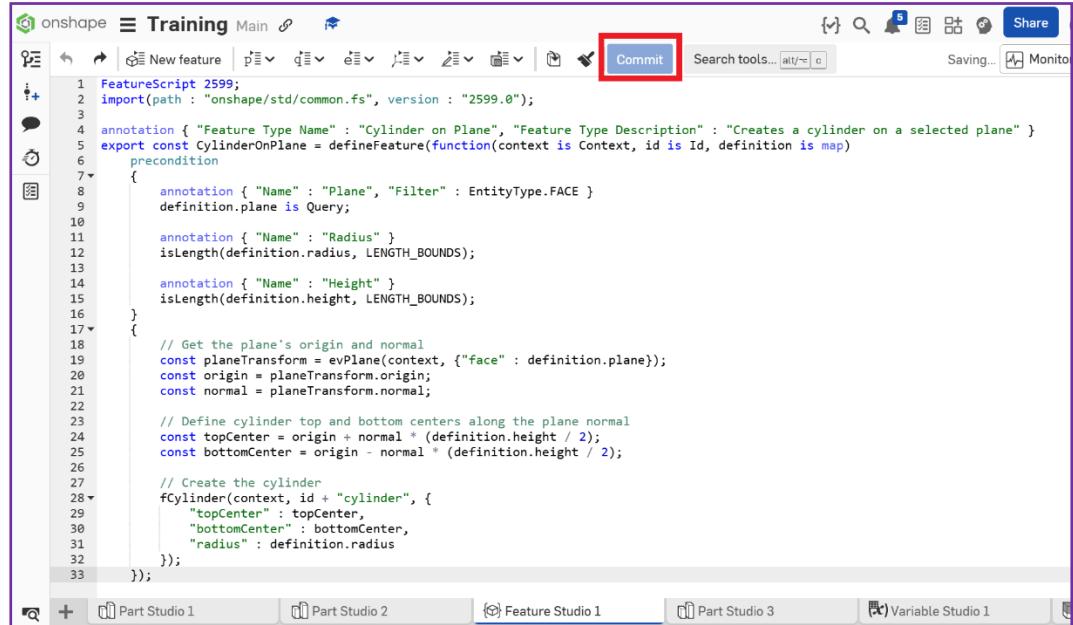
 6. The Feature Studio codes can also be generated using ChatGPT. However, keep in mind that the available data is limited, and the codes may require a few iterations of corrections to run without errors.
- Follow these steps to generate Feature Studio code using ChatGPT:**
- Copy the code from [**Onshape_Featurescript_Samplecode_Cylinder.txt**](#)
 - Start a new chat in ChatGPT, paste the code into the prompt, and add "Learn this code" (e.g., {Code} - Learn this code).

```
FeatureScript 2599;
import(path : "onshape/std/common.fs", version : "2599.0");

annotation { "Feature Type Name" : "Test1", "Feature Type
Description" : "Creates a simple cylinder" }
export const Test1 = defineFeature(function(context is Context, id
is Id, definition is map)
    precondition
    {
        annotation { "Name" : "Radius" }
        isLength(definition.radius, LENGTH_BOUNDS);
    }
    {
        // Create a cylinder at the origin
        fCylinder(context, id + "cylinder1", {
            "topCenter" : vector(0, 0, 0) * inch,
            "bottomCenter" : vector(1, 1, 1) * inch,
            "radius" : definition.radius
        });
    };
```

{ - Learn this code }

- Enter the prompt: "**Modify the Feature Script code to generate the cylinder on a selected plane.**"
- Copy the updated code from ChatGPT, paste it into the Onshape Feature Studio editor, and commit the changes.



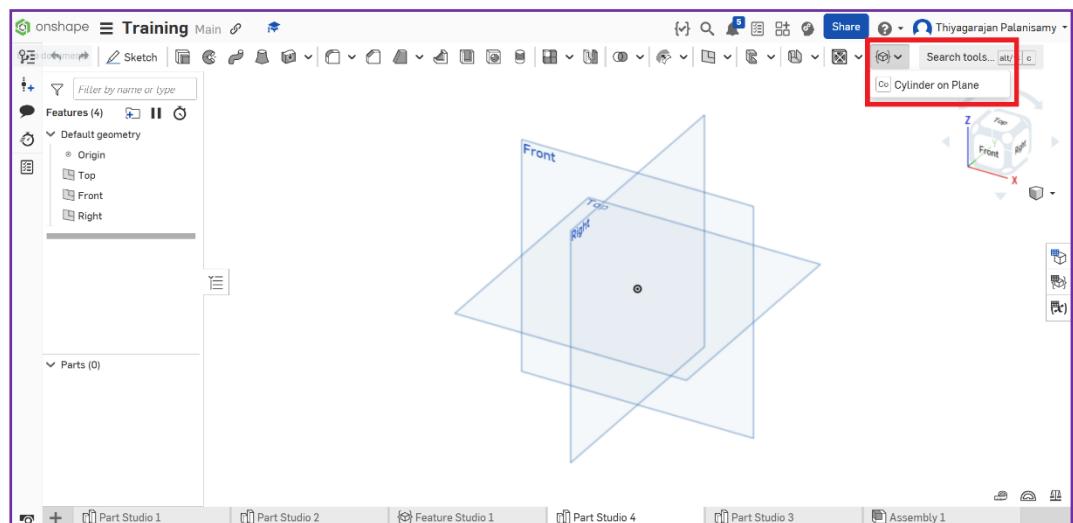
```

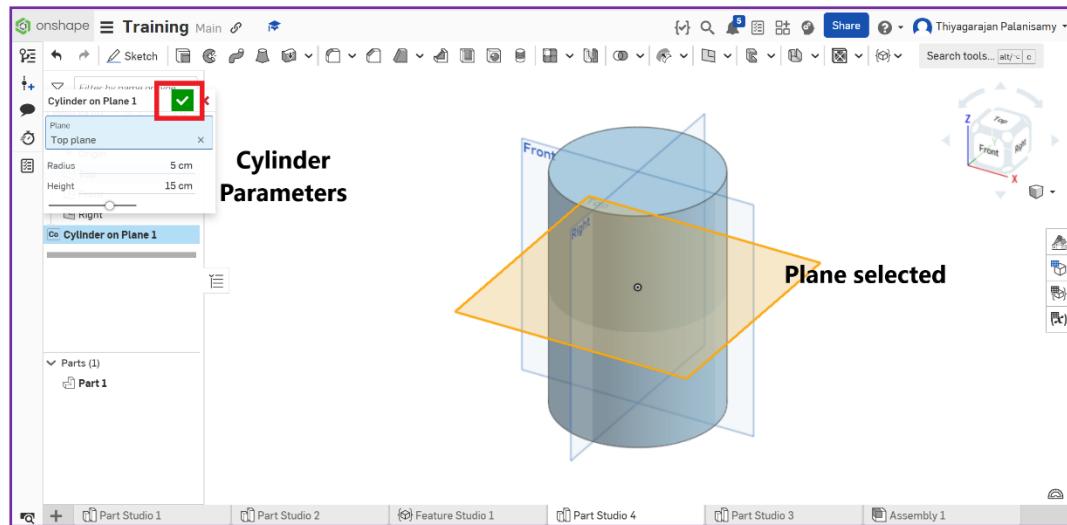
onshape Training Main ⌂
New feature | ⌂ v ⌂ v ⌂ v ⌂ v ⌂ v ⌂ v ⌂ v ⌂ v ⌂ v ⌂ v Commit Search tools... Saving... Monitor
1 FeatureScript 2599;
2 import(path : "onshape/std/common.fs", version : "2599.0");
3
4 annotation { "Feature Type Name" : "Cylinder on Plane", "Feature Type Description" : "Creates a cylinder on a selected plane" }
5 export const CylinderOnPlane = defineFeature(function(context is Context, id is Id, definition is map)
6   precondition
7   {
8     annotation { "Name" : "Plane", "Filter" : EntityType.FACE }
9     definition.plane is Query;
10
11     annotation { "Name" : "Radius" }
12     isLength(definition.radius, LENGTH_BOUNDS);
13
14     annotation { "Name" : "Height" }
15     isLength(definition.height, LENGTH_BOUNDS);
16   }
17
18   // Get the plane's origin and normal
19   const planeTransform = evPlane(context, { "face" : definition.plane });
20   const origin = planeTransform.origin;
21   const normal = planeTransform.normal;
22
23   // Define cylinder top and bottom centers along the plane normal
24   const topCenter = origin + normal * (definition.height / 2);
25   const bottomCenter = origin - normal * (definition.height / 2);
26
27   // Create the cylinder
28   fcylinder(context, id + "cylinder", {
29     "topCenter" : topCenter,
30     "bottomCenter" : bottomCenter,
31     "radius" : definition.radius
32   });
33 });

```

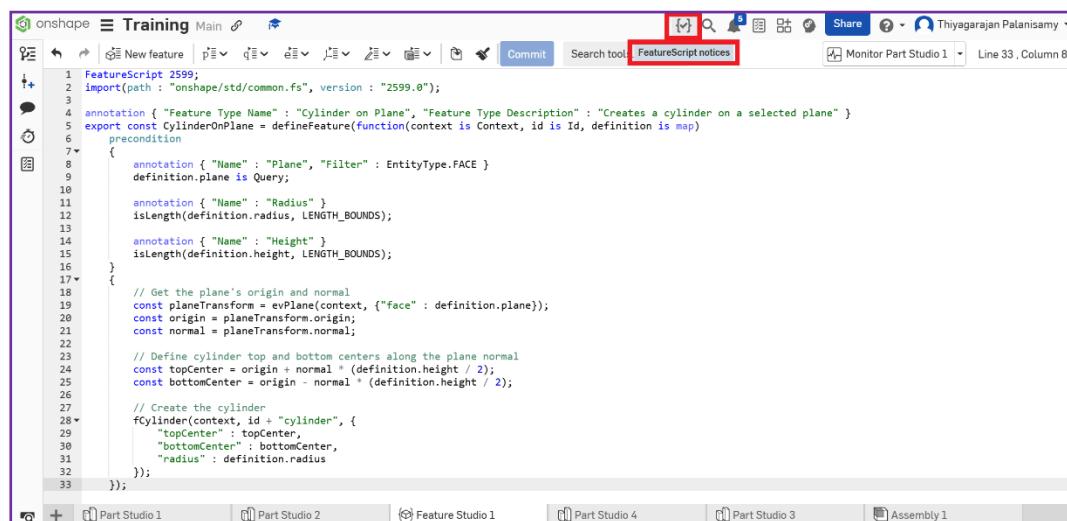
The screenshot shows the Onshape Feature Studio interface. The main area displays the FeatureScript code for creating a cylinder on a selected plane. The 'Commit' button at the top right is highlighted with a red box. Below the code, there are tabs for Part Studio 1, Part Studio 2, Feature Studio 1, Part Studio 3, and Variable Studio 1. The Feature Studio 1 tab is active.

- Create a new Part Studio tab and select **Cylinder on Plane** Feature to verify if the code functions correctly.





- Use the “FeatureScript notices” option for code debugging. It helps identify errors and warnings in the Feature Studio script, making troubleshooting easier.

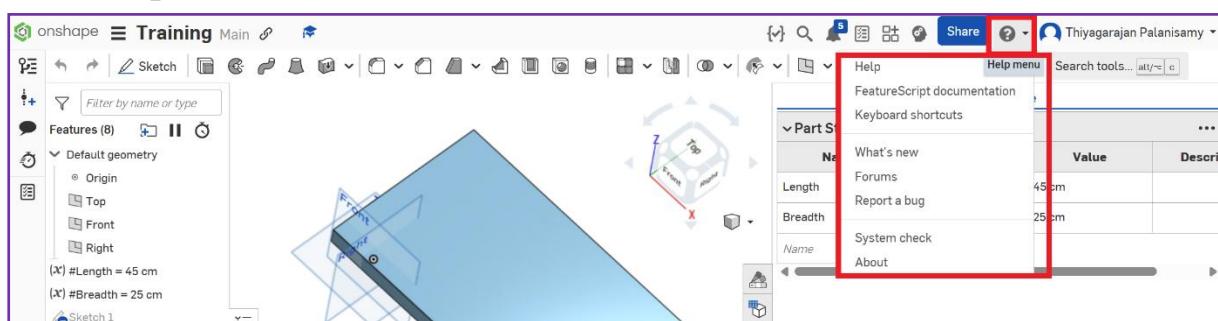


```

1 FeatureScript 2599;
2 import(path : "onshape/std/common.fs", version : "2599.0");
3
4 annotation { "Name" : "Plane", "Filter" : EntityType.FACE };
5 export const CylinderOnPlane = defineFeature(function(context, id, definition) {
6   precondition
7   {
8     annotation { "Name" : "Plane", "Filter" : Query };
9     definition.plane = Query;
10
11     annotation { "Name" : "Radius" };
12     isLength(definition.radius, LENGTH_BOUNDS);
13
14     annotation { "Name" : "Height" };
15     isLength(definition.height, LENGTH_BOUNDS);
16   }
17
18   // Get the plane's origin and normal
19   const planeTransform = evPlane(context, { "face" : definition.plane });
20   const origin = planeTransform.origin;
21   const normal = planeTransform.normal;
22
23   // Define cylinder top and bottom centers along the plane normal
24   const topCenter = origin + normal * (definition.height / 2);
25   const bottomCenter = origin - normal * (definition.height / 2);
26
27   // Create the cylinder
28   fCylinder(context, id + "cylinder", {
29     "topCenter" : topCenter,
30     "bottomCenter" : bottomCenter,
31     "radius" : definition.radius
32   });
33 });

```

7. Click on the “Help” menu to access **Feature Documentation**, **Shortcuts**, **Onshape version**, and other useful resources.



8. Onshape FeatureScript Library containing source codes

<https://cad.onshape.com/documents/12312312345abcabcabcdeff/w/a855e4161c814f2e9ab3698a/e/daec16bcbf3755b1bc852153>